Mac OS X OpenGL In Depth

Session 201

Mac OS X OpenGL In-Depth

Using OpenGL in the Real World

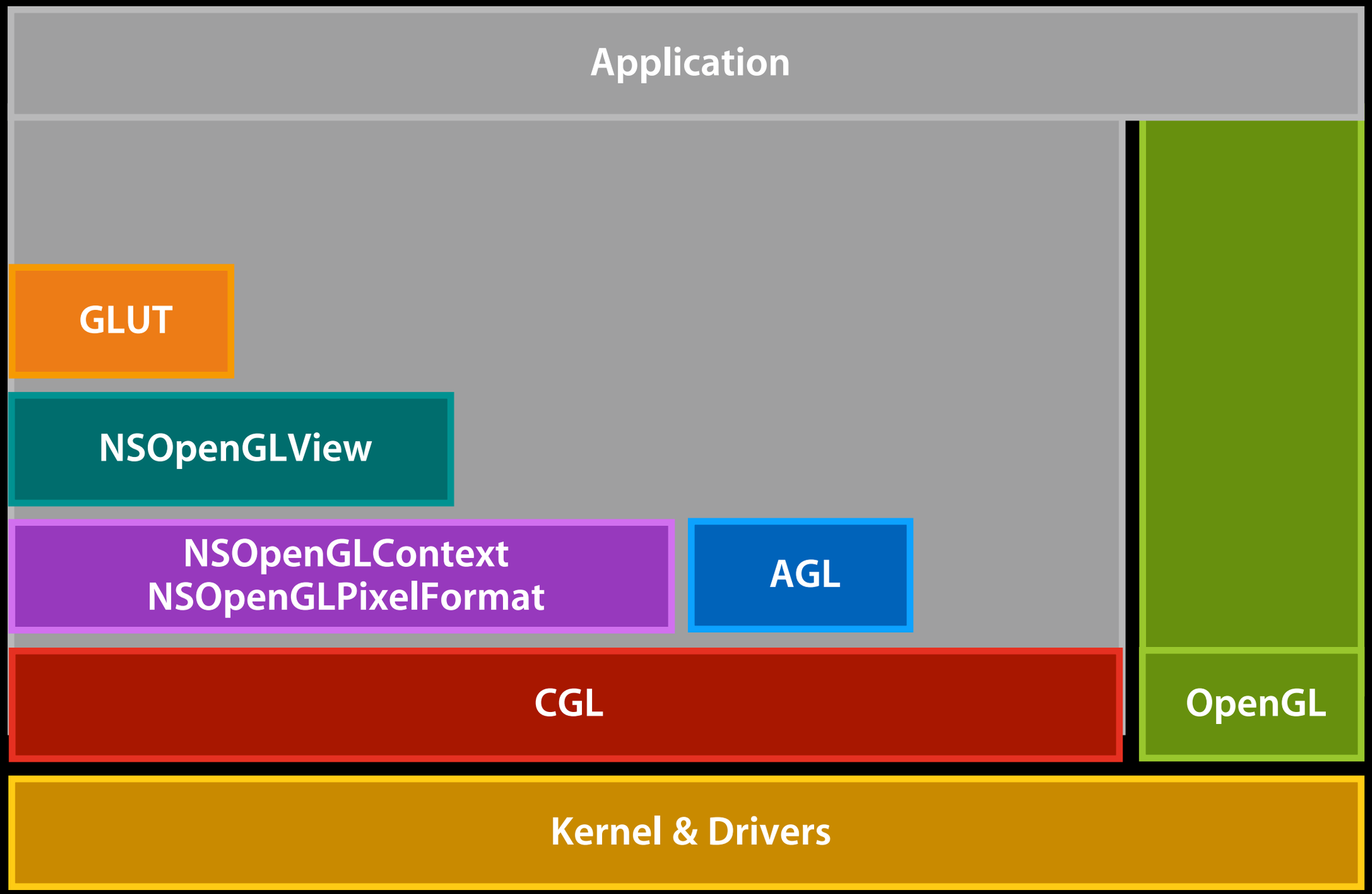
Geoff Stahl
High Level 3D Dude
gstahl@apple.com

Introduction

- Introduce the OpenGL graphics system
- Show how to get started and use OpenGL
- Techniques for using this 3D API in a variety of applications
- Tips for efficient and effective application design

Blue Notes: TN, Q&A and sample references for your use

Technology Framework



Introducing OpenGL

What makes OpenGL appealing?

Introduction

- What is OpenGL?
- Why use OpenGL?
- OpenGL state machine
- API and functionality

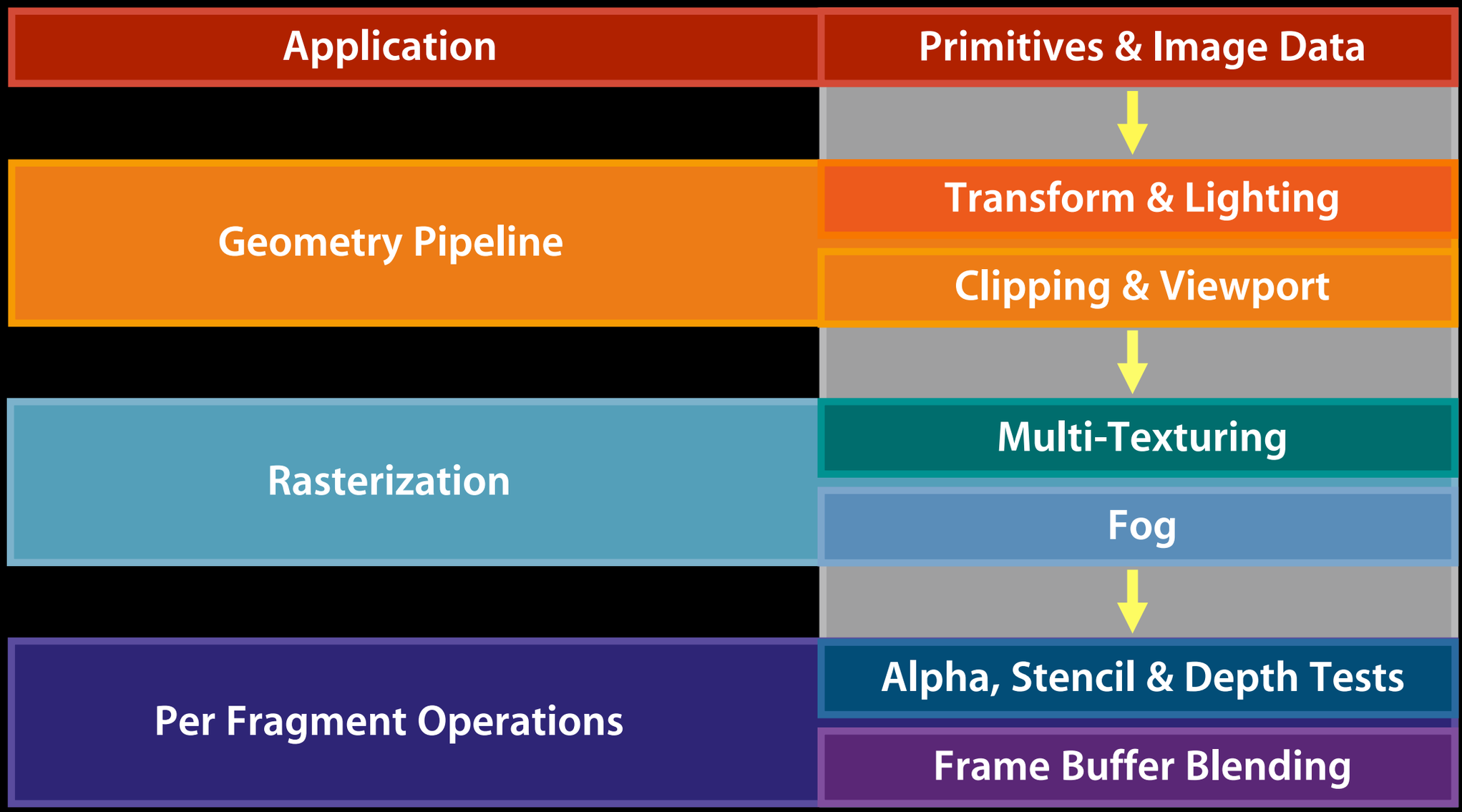
What Is OpenGL?

- Software interface for graphics hardware
  - Platform agnostic
  - Asynchronous
- Geometric primitives
  - Points, lines, polygons
- A state machine
  - Shading, lighting, texture, depth buffering, etc.
  - Projection and model/view transformations

Why Use OpenGL?

- Fast and fully hardware accelerated
  - Measured 650 MB/sec of texture uploaded
    Modifying every pixel in a 4MB texture @ 200Hz+
  - Real-world ability to upload over 140 Mtris/sec
    Sending 100K triangles per frame @ 1,400Hz
  - GPUs with up to 20 GB/sec internal data bandwidth
    Drawing 16MB texture @ 1,200Hz+
  - Uses GPU asynchronously to minimize CPU load
- Large, flexible API
  - Good for both 2D and 3D operations
  - Vertex and pixel programs for complete customization
- Cross platform with numerous code samples

OpenGL Pipeline



OpenGL State Machine

- Understanding is key to successful development
  - States remain until changed
  - State changes possible be expensive
- Examining state
  - Use OpenGL Profiler
- Get and setting state
  - Getting state could involve a pipeline stall

```
glEnable(...)/glDisable(...)
glGet...(...)/glIsEnabled(...)
glPushAttrib(...)/glPopAttrib()
```

API and Functionality

- API
  - Procedural commands using a client-server execution model
  - Setting state, drawing primitives, manipulating buffers
- Functionality
  - OpenGL 1.1 to 1.4
    - Core functionality
  - 80 extensions currently
    - ARB, EXT, APPLE, etc…
    - Extended functionality

TN2080: OpenGL Functionality

Detecting Functionality

- Must have valid context attached to renderer
  - But… Can use CGL directly for this
- Core Version:

```
strVersion = glGetString (GL_VERSION);
```

- Extension Support:
  - See glext.h for extension list

```
strExt = glGetString (GL_EXTENSIONS);
fFence = gluCheckExtension ("GL_APPLE_fence", strExt);
```

- OpenGL Limits:

```
glGetIntegerv (GL_MAX_TEXTURE_SIZE, &maxTextureSize);
```

- Sample code has function to do all this for you!

Sample Code: Carbon OpenGL, Carbon CGL & Cocoa OpenGL

Demo

Carbon OpenGL (glCheck) Sample Code

Interfaces

How to access the capabilities of OpenGL?

Basics

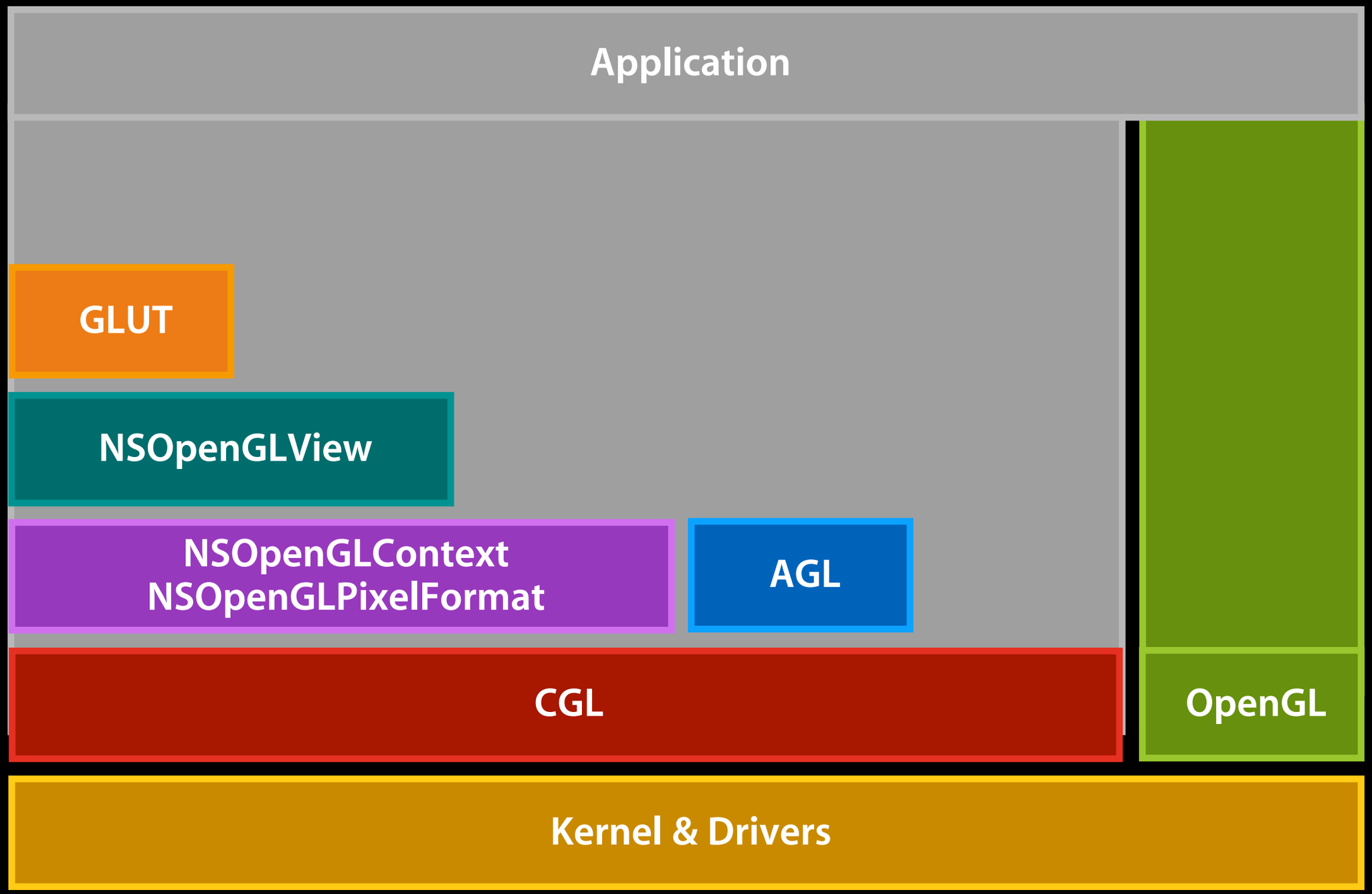
- Pixel format
  - Buffer and capability description
  - Double buffer, depth size, full screen, stencil size, etc.
- Context
  - State bucket
  - Commands are sent to current context
- Drawable
  - Window, view, screen
  - Sets size for context and determines actual renderer
  - Attaching creates buffers

Interfaces

- CGL
  - Low level, full screen only
- AGL
  - Carbon, windowed or full screen
- NSGL
  - Cocoa, view based or custom
- GLUT
  - Cross platform, includes event handling, limited API

Q&A 1160: OpenGL Interfaces

Technology Framework



CGL

- Core OpenGL
  - Low level, basis for AGL and NSGL
  - Full screen only
- Setup
  - Get desired display mode
  - Capture displays
  - Switch to mode
  - Create pixel format for specific display
  - Create context
  - Set full screen

CGL

Code example

```
CGLPixelFormatAttribute attribs[] =
{kCGLPFADisplayMask, 0, kCGLPFAFullScreen,
 kCGLPFADoubleBuffer, kCGLPFADepthSize, 16, NULL};

display = CGMainDisplayID ();
displayMode = CGDisplayBestModeForParameters (display,
                                depth, width, height, NULL);
CGCaptureAllDisplays ();
CGDisplaySwitchToMode (display, displayMode);

attribs[1] = CGDisplayIDToOpenGLDisplayMask (display);
CGLChoosePixelFormat (attribs, &pixelFormat,
                                &numPixelFormats);
CGLCreateContext (pixelFormat, NULL, &cglContext);
CGLDestroyPixelFormat (pixelFormat);
CGLSetCurrentContext (cglContext);
CGLSetFullScreen (cglContext);
```

Sample Code: Carbon CGL

AGL

- Carbon interface
  - Window and full screen support
- Windowed setup
  - Create window (drawable)
  - Create pixel format
    - Note: limits on multi-display systems
  - Create context
  - Attach drawable

Q&A: OGL01 aglChoosePixelFormat, The Inside Scoop
OGL02 Correct Setup of an AGLDrawable

AGL

Code example: windowed

```
GLint attrib[] = {AGL_RGBA, AGL_DOUBLEBUFFER,
AGL_DEPTH_SIZE, 16, AGL_NONE};

aglPixFmt = aglChoosePixelFormat (NULL, 0, attrib);
if (aglPixFmt) {
   aglContext = aglCreateContext (aglPixFmt, NULL);
   if (aglContext) {
      aglSetDrawable (aglContext,GetWindowPort(window));
      aglSetCurrentContext (aglContext);

      // VBL sync
      GLint swap = 1;
      aglSetInteger(aglContext,AGL_SWAP_INTERVAL,&swap);
   }
}
```

Sample Code: Carbon OpenGL

AGL

Code example: full screen

```
GLint attrib[] = {AGL_RGBA, AGL_DOUBLEBUFFER,
AGL_DEPTH_SIZE, 16, AGL_FULL_SCREEN, AGL_NONE};

display = GetMainDevice ();
aglPixFmt = aglChoosePixelFormat (&display, 1, attrib);
if (aglPixFmt) {
   aglContext = aglCreateContext (aglPixFmt, NULL);
   if (aglContext) {
      aglSetFullScreen (aglContext, wdth, hght, frq, 0);
      aglSetCurrentContext (aglContext);

      // VBL sync
      GLint swap = 1;
      aglSetInteger(aglContext,AGL_SWAP_INTERVAL,&swap);
   }
}
```

Sample Code: Carbon AGL Full Screen

AGL

Code example: full screen

```
GLint attrib[] = {AGL_RGBA, AGL_DOUBLEBUFFER,
AGL_DEPTH_SIZE, 16, AGL_FULL_SCREEN, AGL_NONE};

display = GetMainDevice ();
aglPixFmt = aglChoosePixelFormat (&display, 1, attrib);
if (aglPixFmt) {
   aglContext = aglCreateContext (aglPixFmt, NULL);
   if (aglContext) {
      aglSetFullScreen (aglContext, wdth, hght, frq, 0);
      aglSetCurrentContext (aglContext);

      // VBL sync
      GLint swap = 1;
      aglSetInteger(aglContext,AGL_SWAP_INTERVAL,&swap);
   }
}
```

Sample Code: Carbon AGL Full Screen

NSGL

- Cocoa Interface
- NSOpenGLView subclass
  - NSView subclass
  - Contains Single pixel format and context
  - Created in Nib via Interface Builder
- Custom NSView subclass
  - Special cases
    - Multiple contexts or pixel formats
  - See sample for more information

Sample Code: Cocoa OpenGL & Custom Cocoa OpenGL

NSOpenGLView

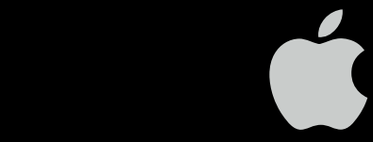
- Interface Builder
  - Use “Custom View” or “OpenGLView”
- -initWithFrame or -initWithCoder
  - Set up pixel format
- -prepareOpenGL (new for Panther)
  - Initialize GL state
  - Or use conditional call from -drawRect
- -reshape
  - Not recommended nor needed
  - Respond to size changes in -drawRect

Sample Code: Cocoa OpenGL

NSOpenGLView (Cont.)

- -update
  - Override not required
    - If overriding, call [super update] first
- -drawRect
  - Handle resize, projection updates
  - Draw content
- Animation
  - NSTimer
    - Call drawRect directly
    - NSDefaultRunLoopMode
    - NSEventTrackingRunLoopMode

QA1209: Updating OpenGL Contexts
Sample Code: Cocoa OpenGL



Demo

Creating an NSOpenGL App

GLUT

- Limited yet simple API
  - Callback based
- Setup
  - Initialize
  - Create Window(s)
  - Set Callbacks
    - Not all callbacks are per window
  - glutMainLoop
    - Do not expect to exit

GLUT

Code Example: Main

```
glutInit(&argc, (char **)argv);
glutInitDisplayMode(GLUT_DOUBLE|GLUT_RGB|GLUT_DEPTH);
glutInitWindowSize (800, 600);
gMainWindow = glutCreateWindow("GLUT");

initOpenGLState(); // standard OpenGL init

glutReshapeFunc (reshape);        // resize
glutDisplayFunc (maindisplay);    // draw
glutKeyboardFunc (key);           // standard key down
glutMouseFunc (mouse);            // mouse button
glutIdleFunc (animate);           // repeating idle

glutMainLoop();
```

Sample Code: GLUT Basics

Techniques

How to harness OpenGL's power?

Techniques

- Textures
- Fonts
- Images
- Movies

Texturing
Carbon using QuickTime

- Texture from packed 8888 or 1555 ARGB buffer
  - Allocate texture sized buffer (NewPtr, malloc, etc.)
  - Create GWorld from buffer with QTNewGWorldFromPtr
  - Draw in to GWorld
  - Dispose GWorld (if desired)
  - Texture from buffer
    - GL_APPLE_packed_pixels extension (all Mac OS X)
      - GL_UNSIGNED_INT_8_8_8_8_REV or GL_UNSIGNED_SHORT_1_5_5_5_REV

Texturing

Code Example: Carbon

```
GetGraphicsImporterForFile(&fsspecImage, &giComp);
GraphicsImportGetNaturalBounds(giComp, &rImage);
hImageDesc =(ImageDescriptionHandle) NewHandle
                        (sizeof (ImageDescriptionHandle));
HLock((Handle) hIDesc);
GraphicsImportGetImageDescription(giComp, &hIDesc);
depth = (**hIDesc).depth;
height = rImage.bottom - rImage.top;
width = rImage.right - rImage.left;
stride = width * depth / 8;
pBuffer = (unsigned char *)NewPtrClear(height * stride);
QTNewGWorldFromPtr(&pGWorld,
  depth == 16 ? k16BE555PixelFormat : k32ARGBPixelFormat,
  &rImage, NULL, NULL, 0, pBuffer, stride);
```

Sample Code: OpenGL Image

Texturing

Code Example: Carbon (Cont.)

```
GraphicsImportSetGWorld(giComp, pGWorld, NULL);
GraphicsImportSetQuality(giComp, codecLosslessQuality);
if (GetGWorldPixMap(pGWorld) &&
    LockPixels(GetGWorldPixMap (pGWorld)))
    GraphicsImportDraw(giComp);                    // draw here
UnlockPixels(GetGWorldPixMap (pGWorld));
CloseComponent(giComp);
DisposeGWorld (pGWorld);

glTexImage2D(GL_TEXTURE_RECTANGLE_EXT, 0, GL_RGBA,
             width, height, 0, GL_BGRA_EXT,
             depth == 16 ? GL_UNSIGNED_SHORT_1_5_5_5_REV:
                           GL_UNSIGNED_INT_8_8_8_8_REV,
             pBuffer);
```

Sample Code: OpenGL Image

Texturing

Cocoa using NSImage and NSBitmapImageRep

- Texture from packed 8888 RGBA buffer
  - Create texture sized NSImage
  - [image lockFocus] and draw
  - Create NSBitmapImageRep from NSImage
    - Texture size is [bitmap size]
  - Texture from [bitmap bitmapData]
    - GL_RGBA and GL_UNSIGNED_BYTE

Texturing

Code Example: Cocoa

```
image = [[NSImage alloc] initWithSize:frameSize];
[image lockFocus];
// draw here
bitmap = [[NSBitmapImageRep alloc]
         initWithFocusedViewRect:NSMakeRect (0.0f, 0.0f,
                        frameSize.width, frameSize.height)];
[image unlockFocus];
texSize.width = [bitmap size].width;
texSize.height = [bitmap size].height;
glTexImage2D (GL_TEXTURE_RECTANGLE_EXT, 0, GL_RGBA,
                texSize.width, texSize.height, 0,
                GL_RGBA, GL_UNSIGNED_BYTE,
                [bitmap bitmapData]);
[bitmap release];
[image release];
```

Sample Code: Cocoa OpenGL

Drawing Fonts

- Per character:
  - aglUseFont
    - Bitmap based, not high performance
- Per string:
  - Draw string to image
  - Texture from image
  - Tips
    - Limit updates
    - Use pre-multiplied alpha
    - Use polygon for transparency and colorizing

Sidebar…

Premultiplied Alpha

- What is it?
  - Alpha which is multiplied through each color component
- Why use it?
  - Simpler compositing formula with closed **over** operator

    $C' = bB + aA - baA$

    $C' = B' + A' - bA'$
  - Different OpenGL Blending

```
// non-premultiplied case
glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA)

// premultiplied case
glBlendFunc(GL_ONE, GL_ONE_MINUS_SRC_ALPHA)
```

Drawing Fonts

- Per character:
  - Quartz 2D Extreme
- Per string:
  - Draw string to image
  - Texture from image
  - Tips
    - Limit updates
    - Use pre-multiplied alpha
    - Use polygon for transparency and colorizing

Demo

Cocoa OpenGL Sample Code

Handling Images

- Display
  - Orthographic projection
  - Scale polygon as appropriate
- Texturing
  - Scale to power of 2
  - Segment to power of 2 rectangles
  - GL_EXT_texture_rectangle extension
    - Non-normalized texture coordinates

Sample Code: OpenGL Image

Demo

OpenGL Image Sample Code

Playing Movies

Using QuickTime

- Setup
  - Standard QuickTime techniques
- Synchronization
  - NewMovieDrawingCompleteUPP
  - Partial sync
  - Full sync requires custom codec
  - See sample
- Drawing
  - Initial glTexImage2D then glTexSubImage2D for updates

Sample Code: OpenGL Movie & OpenGLCompositorLab

Demo

OpenGL Movie Sample Code

Tips

How to add polish and performance to OpenGL?

Tips

- Shared Contexts
- Full Scene Anti-aliasing
- Render to texture

Shared Contexts

- Share object resources and state
  - Texture objects, vertex programs, fragment programs, display lists, vertex array objects
  - Context state not shared
- Same virtual screen configuration
  - Create single display pixel format
  - Share with other single same display pixels formats
- Full screen pixel formats and windowed drawables
  - Will render on specified renderer
  - New for Panther

QA 1248: Context Sharing Tips

Shared Contexts

Code Example

```
GLint attrib[] = {AGL_RGBA, AGL_DOUBLEBUFFER,
AGL_FULL_SCREEN, AGL_NONE};

GLint attrib2[] = {AGL_RGBA, AGL_DOUBLEBUFFER,
AGL_NONE};

disp = GetMainDevice ();

aglPixFmt = aglChoosePixelFormat (&disp, 1, attrib);
aglContext = aglCreateContext (aglPixFmt, NULL);

// use same display
aglPixFmt2 = aglChoosePixelFormat (&disp, 1, attrib2);
aglContext2 = aglCreateContext (aglPixFmt, aglContext);
```

Full Scene Anti-Aliasing

- Exposed via ARB_multisample extension
  - See extensions for details
- Setup
  - Create pixel format with both samples and sample buffers
    - 1 sample buffer and either 2 or 4 samples supported
    - Increased buffer memory
  - Enable MULTISAMPLE_ARB
  - Optional
    - Set hint MULTISAMPLE_FILTER_HINT_NV
      - NICEST or FASTEST

QA 1268: Sharpening FSAA Details

Full Scene Anti-Aliasing

Code Example

```
GLint attrib[] = { AGL_SAMPLE_BUFFERS_ARB, 1,
AGL_SAMPLES_ARB, 4, AGL_NO_RECOVERY, AGL_NONE };

aglPixFmt = aglChoosePixelFormat (NULL, 0, attrib);
if (aglPixFmt) {
   aglContext = aglCreateContext (aglPixFmt, NULL);
   if (aglContext) {
      aglSetDrawable (aglCntx,GetWindowPort(window));
      aglSetCurrentContext (pWindowInfo->aglContext);

      glEnable (GL_MULTISAMPLE_ARB);
      glHint (GL_MULTISAMPLE_FILTER_HINT_NV, GL_NICEST);
   }
}
```

Sample Code: Carbon OpenGL & Carbon CGL

Full Scene Anti-Aliasing

Code Example

```
GLint attrib[] = { AGL_SAMPLE_BUFFERS_ARB, 1,
AGL_SAMPLES_ARB, 4, AGL_NO_RECOVERY, AGL_NONE };

aglPixFmt = aglChoosePixelFormat (NULL, 0, attrib);
if (aglPixFmt) {
   aglContext = aglCreateContext (aglPixFmt, NULL);
   if (aglContext) {
      aglSetDrawable (aglCntx,GetWindowPort(window));
      aglSetCurrentContext (pWindowInfo->aglContext);

      glEnable (GL_MULTISAMPLE_ARB);
      glHint (GL_MULTISAMPLE_FILTER_HINT_NV, GL_NICEST);
   }
}
```

Sample Code: Carbon OpenGL & Carbon CGL

Render To Texture

- Surface texture
  - aglSurfaceTexture, glutSurfaceTexture, -createTexture: fromView: internalFormat:
  - For rendering to surface and texturing from surface
- Pbuffers
  - Corresponds to WGL Pbuffers
  - Simplified set up
  - More robust than surface textures
  - AGL, CGL, NSGL (soon) support
- SuperBuffers
  - Working with ARB toward final specification

Sample Code: Carbon Pbuffer, AGLSurfaceTexture & GLUTSurfaceTexture

Pbuffers

- Generalized Pixel Buffers
  - Target of rendering
    - aglSetPBuffer, CGLSetPBuffer
  - Source for texturing
    - aglTexImagePBuffer, CGLTexImagePBuffer
  - Hardware accelerated
- Setup
  - Create
  - Update (draw to)
  - Bind (texture from)
  - Destroy

Pbuffers

Code Example: Carbon

```
 // create Pbuffer
GLint attrib[] = {AGL_RGBA, AGL_RED_SIZE, 8,
                  AGL_ALPHA_SIZE, 8, AGL_DEPTH_SIZE, 24,
                  AGL_CLOSEST_POLICY, AGL_NONE };
 aglPixFmt = aglChoosePixelFormat (NULL, 0, attrib);
 aglContext = aglCreateContext (aglPixFmt, NULL);
 aglCreatePBuffer (kPbufferWidth, kPbufferHeight,
                   GL_TEXTURE_2D, GL_RGBA,
                   kPbufferMaxLevels, &aglPbuffer)
// update Pbuffer
aglSetCurrentContext (aglContext);
aglSetPBuffer (aglContext, aglPbuffer, 0, 0,
               aglGetVirtualScreen (targetContext));
   // draw here
glFlush ();
aglSetCurrentContext (NULL); // not drawing to Pbuffer
```

Sample Code: Carbon Pbuffer

Pbuffers

Code Example: Carbon (Cont.)

```
// texture with Pbuffer
if (0 == texName) {
   // Note: must call TexImage to pick up mipmap changes
   glGenTextures (1, &texName);
   glBindTexture (GL_TEXTURE_2D, texName);
   glTexParameterf (GL_TEXTURE_2D,
                    GL_TEXTURE_MIN_FILTER, GL_LINEAR);
   aglTexImagePBuffer (targetContext, aglPbuffer,
                       GL_FRONT);
} else // Note: use texture name for Pbuffer type only
   glBindTexture (GL_TEXTURE_2D, texName);
// destroy Pbuffer
glDeleteTextures (1, &texName);
aglDestroyPBuffer (aglPbuffer);
aglDestroyContext (aglContext);
aglDestroyPixelFormat (aglPixFmt);
```

Sample Code: Carbon Pbuffer

Demo

Carbon Pbuffer Sample Code

Wrap Up

Wrap Up

- Introduction
- Interfaces
- Techniques
- Tips

For More Information

Reference Library

Web Sites

Apple OpenGL Developer Site
http://developer.apple.com/opengl

OpenGL.org Site
http://www.opengl.org

Reference Library

Documentation

OpenGL
CGL
NSGL
AGL

See documentation link on Apple OpenGL developer site

http://developer.apple.com/wwdc2003/urls.html

Reference Library

Technical Notes

TN2080: OpenGL Functionality

http://developer.apple.com/technotes/

Technical Q&As

OGL01: aglChoosePixelFormat, The Inside Scoop

OGL02: Correct Setup of an AGLDrawable

QA1209: Updating OpenGL Contexts

QA1248: Context Sharing Tips

QA1268: Sharpening FSAA Details

QA1269: Mac OS X OpenGL Interfaces

http://developer.apple.com/qa/

Reference Library

Sample Code

Carbon OpenGL
Carbon CGL
Cocoa OpenGL
Carbon AGL Full Screen
Custom Cocoa OpenGL
GLUT Basics
OpenGL Image
OpenGL Movie
OpenGLCompositorLab
AGLSurfaceTexture
GLUTSurfaceTexture
Carbon Pbuffer

http://developer.apple.com/samplecode/Sample_Code/Graphics_3D.htm

Roadmap

200 Graphics and Imaging Overview
Latest developments in Mac OS X graphics APIs.
Presidio
Tue, 10:30am

201 Mac OS X OpenGL in Depth
Introduction to OpenGL on Mac OS X.
Nob Hill
Tue, 2:00pm

202 Technology Magic of Keynote
How Keynote uses Mac OS X graphics technologies.
Nob Hill
Tue, 3:30pm

204 Image Capture Update
Integrating Image Capture with your product.
Marina
Wed, 9:00am

Roadmap

205 Vertex Programming with OpenGL Super-fast geometry transformations using the GPU.	**Marina** **Wed, 3:30pm**
206 ColorSync Update Latest developments in Mac OS X color management.	**Mission** **Wed, 5:00pm**
207 Quartz 2D in Depth Harnessing the power of Quartz 2D in your product.	**Presidio** **Thu, 9:00am**
208 Fragment Programming with OpenGL Creating incredible per-pixel effects on Mac OS X.	**Pacific Heights** **Thu, 10:30am**

Roadmap

209 OpenGL Optimizations Obtaining maximum 3D performance on Mac OS X.	**Marina** **Thu, 2:00pm**
210 Mac OS X Printing Update Latest developments in the Mac OS X printing system.	**Mission** **Thu, 3:30pm**
211 Introduction to Quartz Services Capturing & configuring displays on Mac OS X.	**Pacific Heights** **Fri, 9:00am**
212 Cutting Edge OpenGL Techniques Expert recipes for stunning 3D effects.	**Russian Hill** **Fri, 2:00pm**

Roadmap

421 Mac OS X Accessibility
Latest developments in Mac OS X accessibility.

Presidio
Fri, 10:30am

FF018 Graphics & Imaging Feedback
Share your thoughts on graphics & imaging technologies.

North Beach
Fri, 5:00pm

Who to Contact

Geoff Stahl
High Level 3D Dude
gstahl@apple.com

Travis Brown
Graphics & Imaging Evanglist
email@apple.com

http://developer.apple.com/wwdc2003/urls.html

Q&A

Travis W. Brown
Graphics & Imaging Evangelist
Worldwide Developer Relations

http://developer.apple.com/wwdc2003/urls.html

WWDC 2003

Worldwide Developers Conference